# java学习基础

## 一．基本知识

1. 对象

万物皆对象，客观存在的事物都是对象

1. 面向对象

人　关注　>具体事物

1. 类

类是模子，确定对象将会拥有的特征（属性）和行为（方法）

类的特点：

类是对象的类型

具有相同属性和方法的一组对象的集合

1. 对象的属性

　属性：对象具有的各种特征值

　每个对象的每个属性都拥有特定值

1. 对象的方法

方法：对象执行的操作

1. 类和对象的关系

　类是一个抽象的概念，仅仅是模板

对象是一个实际存在的事物

1. 定义类

泪的重要性：所有java程序都以类class为组织单元

定义类的步骤：

a.定义类名

example1:

public class 类名｛

//定义属性部分

//定义类的方法

｝

1. 对象

　a:创建对象

　　类名　　对象名＝new 类名();

　b:使用对象

1. 成员变量和局部变量

　　成员变量：

　　在类中定义，用来描述对象将要有什么

　　局部变量：

　　　　在类的方法中定义，在方法中临时保存数据

区别：

　作用域不同：

　　　局部变量的作用域仅限于定义的方法中

　　　成员变量的作用域在整个类内部都是可见的

　初始值不同：

成员变量初始值会默认为0

局部变量不会被初始化（必须人为赋初值，不赋初始值，程序编译会出错）

两类重名时：

　　会优先使用局部变量

1. 构造方法

　Telephone phone=new Telephone();

使用new+构造方法 创建一个新的对象

构造方法是定义在java类中的一个用来初始化对象的方法

　构造方法没有返回值，与类同名　。当没有指定构造方法时，系统默认会有一个不带参数的构造方法。

当自定义有参的构造方法（系统便不会再生成无参的构造方法），并且没有自定义无参的构造方法，则不能使用无参的构造方法。

1. static使用之静态常量

　　java中被static修饰的成员称为静态成员或类成员。它属于整个类所有，而不是某个对象所有，即被类的所有对象所共享。静态成员可以使用类名直接访问。

1. static使用之静态方法

使用static修饰的方法，称为静态方法

　a．静态方法中可以直接使用同类中的静态成员，但不能直接使用费静态成员（需要通过对象来访问)

　b.在普通成员方法中，则可以直接使用静态变量和费静态变量

　c.静态方法中不能直接访问非静态方法，需要通过对象来访问静态方法

1. 静态初始化块

　只在类加载时执行，且只会执行一次，同时静态初始化块只能给静态变量赋值，不能初始化普通的成员变量

## 二．特殊性质

　１．封装

　将类的某些信息隐藏在类内部，不允许外部程序直接访问，而是通过该类提供的方法来实现对隐藏信息的操作和访问。

　好处：只能通过规定的方法访问数据

　　　封装内部实现细节

　（１）封装的实现步骤

修改属性的可见性

创建getter/setter方法

在getter/setter方法中加入属性控制语句

1. 使用java中的包

　包的作用：

　　　解决java文件

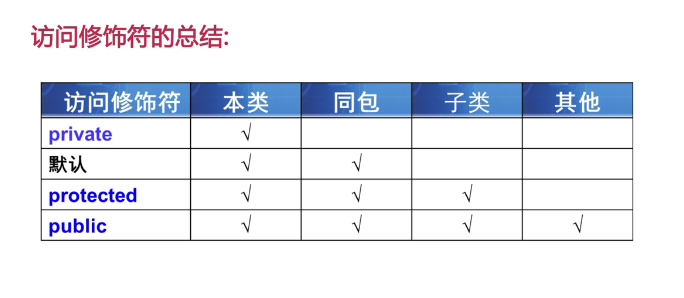
　　　解决同名文件冲突

　包的使用：

　　包的命名使用小写的英文字母

　　使用import 包名　导入包名

　（３）java中的访问修饰符



(4)this

　this关键字代表当前对象

　　　this.属性　　操作当前对象的属性

　　　this.方法　　调用当前对象的方法

封装对象的属性的时候，经常会使用this关键字

1. 内部类

　　　　定义在一个类里面的类。与之对应，包含内部类的类，称为外部类。

　　　内部类的主要作用：

　　　　　　　提供了更好的封装，将内部类隐藏在外部类之内

　　　　　　　内部类的方法可以直接访问外部类的所有数据

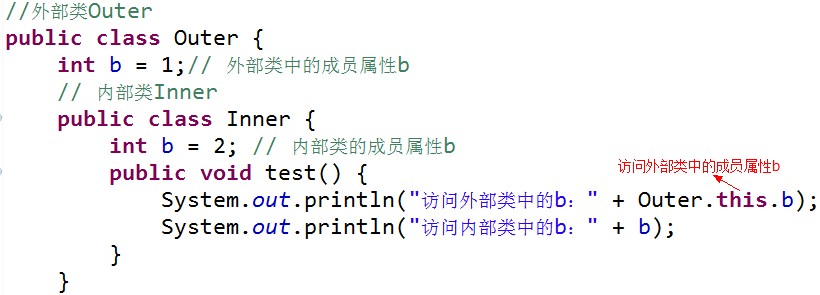
　　　　　a:成员内部类

　　　　　　　定义了成员内部类后，必须使用外部类对象来创建内部类对象

　　　　　　　　　内部类　　对象名＝外部类对象.new 内部类();

外部类不能直接使用内部类的成员和方法（如果需要使用，先创建内部类，通过内部类对象来访问其成员变量和方法）

　　　　　　　　　外部类和内部类具有相同的成员变量或方法，内部类默认使用自己的成员变量和方法，如果需要访问外部类的成员变量，可以使用this关键字



b.静态内部类

　　　静态内部类不能再直接访问外部类的非静态成员，但可以通过new 外部类().成员的方式访问

外部类的静态成员与内部类的成员名称相同，可通过　类名.静态成员　的方式访问外部类的静态成员。如果外部类静态成员与内部类不相同，则可以通过“成员名”直接调用外部类的静态成员

创建静态内部类的对象时，不需要外部类的对象，可以直接创建　内部类　对象名＝new 内部类();

c:方法内部类

　　方法内部类是定义在外部类的方法中，方法内部类只在该方法的内部可见，即只在该方法内可以使用

　　！！！注意：方法内部类不能再外部类的方法以外的地方使用，因此方法内部类不能使用访问控制符和static修饰符

1. 继承

（１）继承的概念

　　　　　继承是类与类的一种关系，是一种is a　关系

1. 继承的好处：

　　　　子类拥有父类的所有属性和方法（不能是private修饰的）

# （３）语法规则

　　public class dog extend animal{

}

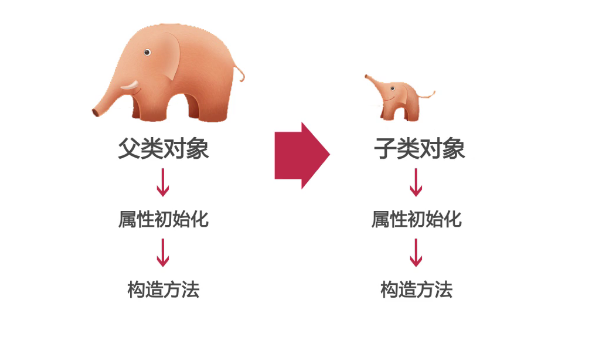
1. 方法的重写

　　　　子类对继承父类的方法不满意，子类重新写父类方法，子类调用时，优先调用子类重写的方法

1. 继承的初始化顺序

初始化父类再初始化子类

先执行初始化对象中的属性，再执行构造方法中的初始化



1. final关键字

使用final关键字做标识有“最终的”的含义

final可以修饰类、方法、属性和变量

　　final修饰类，则该类不允许被继承

　　final修饰方法，则该方法不允许被覆盖

　　final修饰属性　　则该类的属性不会进行隐式的初始化（类的初始化属性必须有值）或者在构造方法中赋值（但只能选其一）

　　final修饰变量，则该变量的值只能赋值一次

1. super关键字

　　　　　在对象的内部使用，可以代表父类对象

　　　　　　访问父类的属性

　　　　　　　　super.age

访问父类的方法

　　　　　　　　super.eat();

子类的构造方法中，必须调用父类的构造方法（隐式显示）

　　　如果子类的构造方法中没有显示调用父类的构造方法，则系统默认调用父类无参的构造方法

　　　如果显示的调用，则必须放在子类构造方法中的第一行

　　　如果子类构造方法中，没有显示调用构造方法。而父类里面有没有无参的构造方法，则会编译出错。

1. object类

　　　　　Object 类是所有类的父类，如果一个类没有使用extends关键字明确标识继承另外一个类，那么这个类默认继承object类

　　　　　object类中的方法，适合所有子类

object类里面的

　toString()方法：

　　　返回的对象的哈希代码（改变返回值，需要重写用　source--->generate toString)

　equals():

　　比较的是对象是否指向同一块内存地址（如果需要比较值是否相同，需要重写）

1. 多态

（１）对象有多种形态

　引用多态

　　　父类的引用可以指向本类的对象

　　　父类的引用可以指向子类的对象

　方法多态

创建本类对象时，调用的方法为本类方法

　创建子类对象时，调用的方法为子类重写的方法或者继承的方法

1. 引用类型转换

　　　　向上类型转换（隐士/自动类型装换），是小类型向大类型的转换

　　　　向下类型转换（强制类型转换，是大类型到小类型转换

　　　　instanceof运算符，来解决引用对象的类型，避免类型转换的安全性问题

（３）抽象类

　　语法定义：抽象类前使用abstract关键字修饰，则该类为抽象类

　　应用场景：在某个情况下，某个父类只是知道其子类应该包含怎样的方法，但无法准确知道这些子类如何实现这些方法

　　　　　　　从多个具有相同特征的类中抽象出一个抽象类，以这个抽象类作为子类的模板，从而避免了子类设计的随意性

　　作用：限制规定子类必须实现某些方法，但不关注实现细节

　　使用规则：

　　　abstract定义抽象类

　　　abstract定义抽象方法，只有声明，不需要实现

　　　包含抽象方法的类是抽象类

　　　抽象类中可以包含普通的方法，也可以没有抽象方法

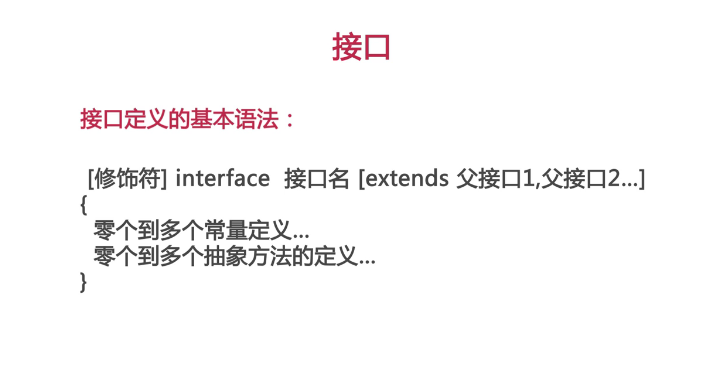
　　　抽象类不能直接创建，可以定义引用变量

1. 接口

　类是一种具体实现体，而接口定义了某一批类所需要遵守的规范，接口不关心这些类的内部数据，也不关心这些类里方法的实现细节，他只规定这些类里必须提供某些方法。

1. 接口定义：

interface（多继承）



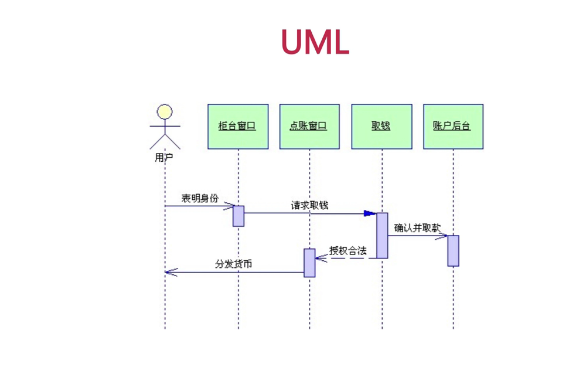
　　　常量：接口中的属性是常量，及时定义时不添加public static final修饰符，系统也会自动加上

接口中方法只能是抽象方法

一个类可以实现一个或多个接口，通过implements



　匿名内部类：没有名字的内部类

UML序列图

项目应用

　哒哒租车系统

1. 数据模型分析
2. 业务模型分析
3. 显示和流程分析